FY16 Hardware Acquisition Planning

Chip Watson

LQCD-ext II Annual Review

May 21, 2015

Outline

The Goal: Maximizing Science

- Historical benchmarking: use of a few key kernels
- Current benchmarking: representative applications
- Software maturity as a constraint

The Probable Contenders

- Conventional x86 cluster
- GPU accelerated cluster
- Xeon Phi / Knights Landing cluster

The Process

- Long range + just-in-time alternatives analysis and optimization
- FY15-FY19 budget constraints & procurement optimizations
- Timeline for the next 12-18 months

Portfolio Optimization

Goal: Optimize the portfolio of machines to get the most science on the portfolio of applications.

A single machine procurement is not driven solely by acquiring the greatest total benchmark suite result (i.e. we don't optimize just one machine).

Instead, for each procurement, the project optimizes its resources to yield the best aggregate performance for its portfolio of applications on its portfolio of hardware.

This allows us to better exploit the trends in the evolution of software, algorithms and hardware, and takes into account which machines are reaching end of life.

Consequently we don't need to have all benchmark applications running on every possible machine type; we can give a zero for that component, and still select that machine if it best optimizes the portfolio.

In recent years, buying a combination of conventional and GPU clusters has produced the best hardware portfolio.

Benchmarking LQCD

For more than a decade, machine performance for LQCD was measured by two key kernels:

- DWF (domain wall fermion) inverter (sparse matrix solver)
- Staggered inverter

These kernels represented a large fraction of the flop/s used in LQCD. With conventional clusters and supercomputers, these kernels were very good predictors of application performance and clock time.

The first iteration of the LQCD Computing project used the average of these two as its benchmark, as a sort of Linpack for LQCD. We continue to track these two to see long range trends.

GPUs forced some changes…

Disruptive Technology

GPUs as inverter accelerators

- Inverters are memory bandwidth intensive, and GPUs have ~6x the memory bandwidth of dual socket conventional x86 servers
- Quad-GPU servers can yield up to a 24x inverter speedup compared to just the host processors

Amdahl's Law constraints

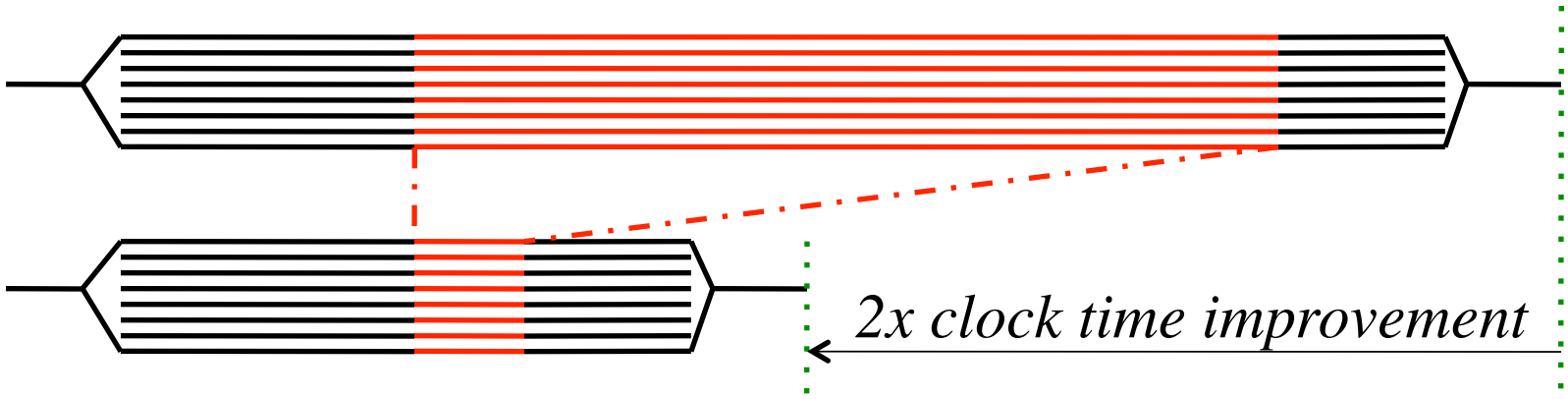
- *Clock time acceleration* is highly dependent upon the fraction of the code's run time spent in the inverter
- The unaccelerated portions of code constrain the overall acceleration.

Amdahl's Law Constraint

A major challenge in exploiting GPUs is Amdahl's Law:

If 60% of the code is GPU accelerated by 6x,

the net gain is only 2x.



Also disappointing in this scenario: the GPU is idle 80% of the time!

Fortunately many LQCD codes spend > 95% of their clock time in a single kernel, a matrix inversion, and so for these applications Amdahl's Law was not (yet) a show-stopper, and gains of 18x using 4 cards were achieved.

Implication: inverter performance is not as good a metric for hardware selection; our process is now is more complicated.

Hand Tweaked Software

Herculean (guru) software efforts can yield high performance gains on the inverters, without making such gains available to the rest of the code base.

=> Amdahl's Law isn't just a GPU phenomena

Xeon Phi / Knight's Landing (KNL) will be self hosted (unlike today's GPU systems) and is likely to have different performance characteristics on different sections of code compared to today's "fat" x86 cores. It will have highly optimized (guru) inverters, but other sections might not perform as well.

This difference between normal compiler performance and the performance of hand tweaked / guru optimized code must be taken into account during alternatives analysis and/or procurement benchmarking.

Effective Performance

The approach we have followed since 2009 to adjust for Amdahl's Law is to use "effective performance" for these accelerated systems, defined as:

the performance of an un-accelerated (set of) node(s) needed to achieve the same clock time on the same ***application***

Thus, if a quad GPU node gives the same performance (application clock time) as a cluster of 16 un-accelerated nodes, then we rate each GPU node for that application as 16x the performance of the un-accelerated node, and we continue to rate the un-accelerated node based upon its inverter performance.

This keeps the same units of "inverter flops", while making the benchmarking process more application oriented.

LQCD Benchmark Suite

Since different applications see different clock time acceleration, the project selects several representative applications to measure real science performance, including the 3 dominant actions in USQCD today: HISQ (highly improved staggered quarks), clover and DWF. The suite includes

- Inverter heavy codes, such as propagator / perambulator generation
- Codes with a larger diversity of operations, including propagator tie-ups and analysis code with CPU-only sections (about 12% of runtime)
- Both mixed-precision and mono-precision (single and double) inverters

Extreme scalability, such as LQCD requires on capability machines for gauge configuration generation, is not a driver in our computing project procurements; the focus is more on analysis and capacity mode running. Even so, job sizes are climbing, and some amount of scalability is always a value.

Current Benchmark Suite

Calculation	Application	Action	Problem Size	precision
propagators	Chroma	clover	48^3 x 128	mixed singl/dbl
f-sub-pi decay	MILC	hisq	48^3 x 64	double
inverter only	QUDA	dwf	32^3 x 64 x Ls=16	

This suite of benchmarks contains all 3 critical actions, with an emphasis on inverters, other linear algebra and general code.

The selection of applications and balance in application types will be updated prior to issuing the RFP so as to reflect the running anticipated in the first year of the new machine.

2016 LQCD Machine

The 5 year plan for FY2015 - FY2019 has leaner budgets, 40% less hardware, with no hardware funds in FY2015, so the project currently plans to minimize labor and maximize hardware by combining funds into two procurements:

- FY16 & FY17 into a 2 phase procurement of ~$1.96M
- FY18 & FY19 into a 2 phase procurement of ~$2.65M

Moving from annual procurements to every other year procurements has reduced anticipated procurement and operating costs by roughly $300K. Further, it allows the possibility of deploying larger homogeneous resources to support occasionally running large jobs.

The project at least annually re-evaluates the split between operations and hardware to optimize science, and in fact every other aspect of the project.

Approximately 6%-8% of the hardware funds are used for file servers, also adjusted as needed to optimize science.

FY16 and FY17 Deployment Goals

The deployed performance goals for FY16 and FY17 are

FY16: 49 TFlops Total: 115 TFlops

FY17: 66 TFlops

These numbers are derived from recent deployments of mixed GPU and conventional resources (typically 40% GPU, 60% conventional by cost, with the performance balance being more like 60% GPU, 40% conventional).

Our ability to deploy the more effective architectures is always constrained by the fraction of our applications able to exploit well their higher performance per dollar.

Multi-Year Funding Approach

The project does not award a purchase order with 2 years of funding. Nor does it do lease-to-own. Instead, it awards with fixed priced options for later procurement. This has multiple benefits:

- It allows delaying the commitment of the final portion of funds until uncertainties are resolved (e.g. next year funding, or optimal split between conventional and GPU)
- It allows for multi-year funding of new capacity with an option to later change the selection of hardware for that second year if something much more cost effective emerges (i.e. something of greater value than a larger homogenous resource)

FY16 Procurement Timeline

July 2015 – Alternatives Analysis & Site Selection

Aug 2015 – Review by Executive Committee

Sept 2015 – FY16 budget finalization

Oct 2015 – Detailed Acquisition Plan

Nov 2015 – RFI

Jan 2016 – Benchmark Suite determination (not yet final code)

Feb 2016 – Benchmarks frozen

Mar 2016 – RFP

May 2016 – Delivery & Commissioning

July 2016 – Operations of 1st half of 2 year procurement

FY17 Expansion Option

July 2016 – Evaluation and Recommendation

- evaluate late emerging alternatives
- if mixed system, can expand either part, or both parts
- re-optimize storage vs. compute

Aug 2016 – Review by Executive Committee

Sept 2016 – FY17 budget finalization

Oct 2016 – Award

Dec 2016 – Delivery & Commissioning (abbreviated)

Jan 2017 – Operations

These are working dates; project milestones for Operations will be set 6 months later to accommodate the *unlikely* event of a continuing resolution.

Hardware Selection

The Probable Contenders:

Conventional x86 Cluster

- ✓ Runs all software at least OK
- ✓ Easily integrated & used
- ✓ Most user friendly for development

NVIDIA Pascal GPU Cluster

- ✓ High flop/s, high memory bandwidth
- ✓ Should run all existing GPU software

Intel Xeon Phi / Knights Landing Cluster

- ✓ High flop/s, high memory bandwidth
- ✓ Might run most software at least OK

(…)

Typical Configurations

Conventional x86

- Dual socket, 16 core Xeon (64 threads), 64 GB memory
- Infiniband, 1:1 QDR or 2:1 FDR

NVIDIA Pascal

- Quad GPU + dual socket CPU (typical: host = 4x-6x GPU memory)
- on package high bandwidth memory
- fatter node therefore higher speed Infiniband, FDR or faster

Intel Xeon Phi (KNL)

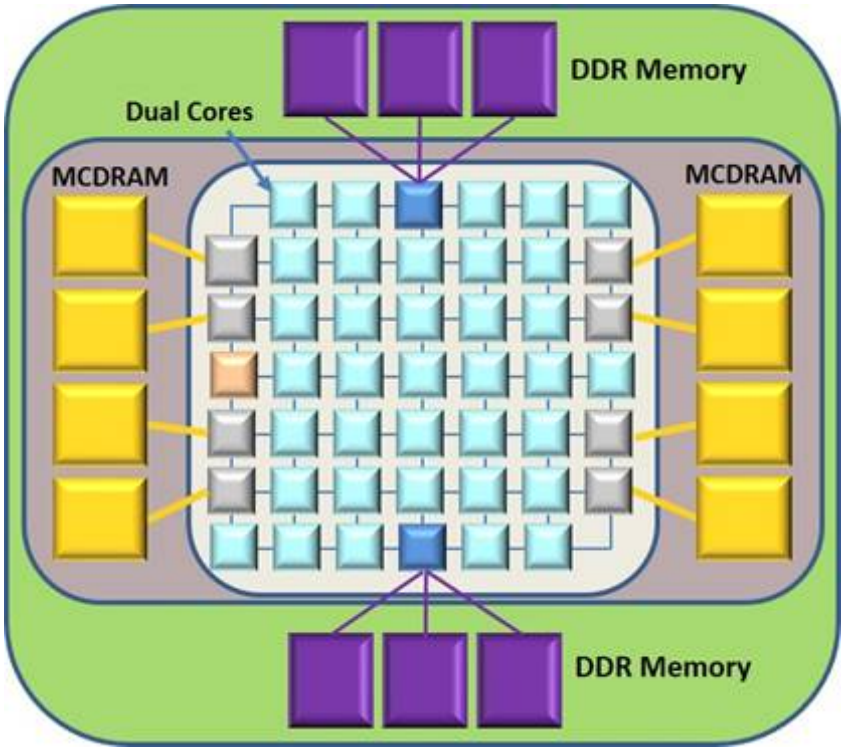
- single socket, 64+ core, 256+ threads, 512 bit SIMD
- 96 GB main memory (6 busses, 48 GB “up to 384 GB”)
- on-package high bandwidth memory “up to 16 GB”

NVIDIA Pascal

- Expected in 2016, details still NDA
- On package “3D Stacked Memory” with “up to 1 TB/s” bandwidth
- NVLink
  - 4 point to point bidirectional 100 Gb/s links per GPU, 5x faster than PCIe
  - Enables tight coupling of 4 GPUs within a node

KNL many core

Not an accelerator. Not a heterogeneous architecture.
x86 single socket node. Better core than KNC:



- Out-of-order execution
- Advanced branch prediction
- Scatter gather
- 8 on package MCDRAMs, "up to 16 GB"
- 6 DDR4 ports "up to 384 GB"
- 1 MB L2 cache per 2 core tile (figure shows up to 72 cores if
- all are real & operational)

https://software.intel.com/en-us/articles/what-disclosures-has-intel-made-about-knights-landing

Considering a New Architecture

USQCD Xeon Phi software maturity is growing

- 2013 saw LQCD running at TACC / Stampede (KNC), with an optimized Dirac inverter matching the performance of a contemporary GPU
- Additional developments under way on multiple codes, driven by large future resources:
  - **Cori**, 2016; with 9,300+ chips,
  - followed by ANL's **Theta** (KNL) in 2016; 2,500 chips
  - and ANL's **Aurora** (KNH – Knights Hill) in 2018, with "50,000 nodes"

Commodity hardware OEMs are planning KNL machines

Conclusion: KNL is viable as an LQCD capacity resource in 2016.

Other Notable Changes Coming

In 2016, both Pascal and Knights Landing should have on package memory – high bandwidth, memory mapped (or cache, but probably better directly managed). Software will need to evolve to exploit this.

At some point, both will have better I/O capabilities:

- Pascal will have a new Nvlink I/O channel. Details still NDA.
- Intel will have an on-chip network that can replace Infiniband, but timeline is still NDA.

The timeline for these is uncertain / NDA. If any improvements are available in time for Phase II of the combined procurement (Fall, 2016), they will be considered, leading to some nodes having additional batch system tags to reflect enhanced capabilities. If such options are known early, attempts will be made to ensure that the FY16 and FY17 hardware could function as a single resource. Vendors will be encouraged to offer such options.

These points are relevant to show that both architectures are still improving and have a future.

Summary

- The project has a long track record of procuring cost optimized resources for LQCD, incorporating multiple resource types to maximize science.
- The process for procurement (including alternatives analysis, benchmarking, and optimizing for LQCD) is mature and well suited to the task.
- Funding is tight, and the project has responded by optimizing operations and procurement costs.
- Next year's procurement will be interesting in that there are multiple viable cluster alternatives.